Towers of Hanoi Recurrence Relation

The Towers of Hanoi problem involves moving n disks from one peg to another, using a third peg as auxiliary storage, under the constraint that no larger disk may be placed on top of a smaller disk. The minimum number of moves required to solve the problem with n disks is known to be $2^n - 1$.

Recurrence Relation

Let $T(n)$ denote the minimum number of moves required to solve the Towers of Hanoi problem with n disks. The recurrence relation is given by:

$$T(n) = 2T(n-1) + 1$$

with the base case:

$$T(0) = 0$$

Recursive Function Analysis

The provided Java method `moves` implements the Towers of Hanoi algorithm recursively:

```
public class TowersOfHanoi
{
    public static void moves(int n, boolean left)
    {
        if (n == 0) return;
        moves(n-1, !left);
        if (left) StdOut.println(n + " left");
        else StdOut.println(n + " right");
        moves(n-1, !left);
    }

    public static void main(String[] args)
    {
        int n = Integer.parseInt(args[0]);
        moves(n, true);
    }
}
```

This function prints the sequence of moves to solve the problem, indicating whether each disk should be moved to the left or right.

Proof by Induction

We will prove by induction that the number of moves made by the recursive function `moves` corresponds to the recurrence relation $T(n) = 2T(n-1) + 1$.

Base Case

For $n = 0$:

- The function `moves(0, true)` immediately returns without making any moves.
- The recurrence relation $T(0) = 0$ also indicates 0 moves are needed.

Thus, the base case holds true.

Inductive Step

Assume that the recurrence relation holds for some integer $k \geq 0$. That is, assume:
$T(k) = 2T(k-1) + 1$

We need to show that $T(k+1) = 2T(k) + 1$.

When the function `moves` is called with $n = k + 1$:

1. The function first makes a recursive call `moves(k, !left)` to move the top k disks to the auxiliary peg. By the inductive hypothesis, this takes $T(k)$ moves.
2. Then, it prints a move to transfer the $(k+1)$-th disk to the target peg, taking 1 additional move.
3. Finally, it makes another recursive call `moves(k, !left)` to move the k disks from the auxiliary peg to the target peg, taking another $T(k)$ moves.

Thus, the total number of moves for $k + 1$ disks is:
$T(k+1) = T(k) + 1 + T(k) = 2T(k) + 1$

This matches the recurrence relation we wanted to prove.

Conclusion

By mathematical induction, we have shown that the minimum possible number of moves needed to solve the Towers of Hanoi problem satisfies the recurrence relation $T(n) = 2T(n-1) + 1$. This is exactly the same as the number of moves used by the provided recursive solution, proving that the recursive method correctly implements the minimum moves required to solve the Towers of Hanoi problem.